REMARKS

Claims 1 through 3, 6, 7 and 9 through 33 are pending in the present application. Claims 4, 5 and 8 were previously canceled. Claims 26 through 33 are newly added.

In section 6 of the Office Action, claims 1 through 3, 6, 7 and 9 through 25 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,549,216 to Schumacher et al. (hereinafter "the Schumacher et al. patent") in view of U.S. Patent No. 6,753,873 to Dixon et al. (hereinafter "the Dixon et al. patent"). Applicants are traversing this rejection.

Claim 1 provides for a process for automatically revising data in a database of file records stored in a computer. The process includes, *inter alia*, (a) identifying an occurrence of an event that occurs while a task is being accomplished to revise an entry in a database of file records, (b) recording in a memory, a response to the event, wherein the response is performed by a human operator interacting with a graphical user interface of a computer, to form one or more emulated responses to the event, wherein the one or more emulated responses are stored in an emulated event handler for performing the task, and (c) executing the task on a selected batch of file records by matching a member of a collection of emulated event handlers to a given event.

The Office Action, on page 3, suggests that the Schumacher et al. patent discloses a task being accomplished to revise an entry in a database of file records. In support of this position, the Office Action states that the Schumacher et al. patent teaches that the occurrence of the events that result from user interaction are to be detected and identified are stored in a file, and the Office Action provides several citations, namely, (a) col. 4, lines 20 – 32, (b) col. 7, lines 15 – 20, (c) col. 1, lines 7 – 15, and (d) the Abstract. Below, Applicants explain that the Schumacher et al. patent does not disclose identifying an occurrence of an event that occurs while a task is being accomplished to revise an entry in a database of file records.

The passage at col. 4, lines 20 – 32 states:

System queue 114 then dispatches events to applet listeners 116. Applet
listeners 116 are objects that detect the occurrence of specified events and
initiate a desired action in response to detecting the event. For example, if a
mouse event listener 116 is placed on a button component of applet 108 and
the user clicks on the button with the mouse, the mouse event listener will
detect the occurrence of the event and will, presumably initiate some action
in response to the event. Similarly, a keyboard event listener placed on
appropriate components of applet 108 will detect user keyboard entries with
respect to those components.

The passage at col. 4, lines 20 – 32 does not even mention revising an entry in a database
of file records. As such, this passage neither discloses nor suggests revising an entry in a
database of file records.

The passage at col. 7, lines 15 – 20 states:

FIG. 5 depicts the process of playing back queued events with applet event
recorder 100 according to one embodiment of the invention. When applet
event recorder 100 loads an automator queue 106 from a from a stored file,
automator 102 pops the queued events off of automator queue 106 in the
same order the events were recorded.

The passage at col. 7, lines 15 – 20 does not even mention revising an entry in a database
of file records. As such, this passage neither discloses nor suggests revising an entry in a
database of file records.

The passage at col. 1, lines 7 – 15 states:

Therefore, there is a need for a simpler method of *testing the Java platform*
that does not require recompilation of code, that does not require the applet
or application to be exited before automation, and that uses fewer system
resources. ...The problems identified above are addressed in the present
invention by method for recording events that result from user interaction
with the various components of an applet, such as a Java applet. (Emphasis
added).

The passage at col. 1, lines 7 – 15 does not even mention revising an entry in a database
of file records. Instead, this passage expressly states that the technique disclosed by the

Schumacher et al. patent is <u>specifically directed toward testing a Java platform</u>. As such, this passage neither discloses nor suggests revising an entry in a database of file records.

The Abstract is more than two hundred words in length, and so, Applicants are not quoting it here. Nevertheless, the Abstract <u>does not even mention revising an entry in a database of file records</u>, and so, neither discloses nor suggests revising an entry in a database of file records.

Applicants wish for the Examiner to note that the Schumacher et al. patent's teaching that the occurrence of an event being detected, identified and stored in a file, merely discloses that an event is stored in a file. It is not a disclosure of any particular activity that occurs while revising an entry in a database of file records. As shown above, **none of the cited passages disclose or suggest revising an entry in a database of file records**. Moreover, the Schumacher et al. patent **does not even mention the term "revise"**. Therefore, Applicants respectfully submit that the Schumacher et al. patent neither discloses nor suggests identifying an occurrence of an event that occurs while a task is being accomplished to revise an entry in a database of file records, as recited in claim 1.

The Dixon et al. patent also does not even mention the term "revise". As such, the Dixon et al. patent does not make up for this deficiency on the part of the Schumacher et al. patent.

The Office Action, on page 4, suggests that the Schumacher et al. patent discloses executing the task on a selected batch of file records by matching a member of a given event. In support of this position, the Office Action states that the Schumacher et al. patent discloses executing the selected events or files with the options shown on FIG. 2, and the Office Action cites several passages, namely (a) col. 6, lines 15 – 35, (b) col. 5, lines 20 – 35, and (c) col. 5, lines 67 – 67 (Applicants assume the Examiner intended to cite col. 5, lines 57 – 67).

The passage at col. 6, lines 15 – 35 explains that when a user performs an action, an event is posted to a system queue. However, it does not explain or suggest how the posted event is employed. Thus, this passage discloses an aspect of <u>recording an event</u>, which is <u>not a disclosure of an employment of an emulated event handler</u>. As such, this passage does not disclose matching a member of a collection of emulated event handlers to a given event.

The passage at col. 5, lines 20 – 35 describes components of the graphical user interface of FIG. 2. Below, Applicants explain that for the graphical user interface of FIG. 2, <u>a user affirmatively loads and runs a recorded event</u>. Hence, <u>there is no need for matching a member of a collection of emulated event handlers to a given event</u>. Thus, this passage does not disclose matching a member of a collection of emulated event handlers to a given event.

The passage at col. 5, lines 57 – 67 describes FIG. 9, which is a flow chart of a method for recording events (col. 5, line 58). Thus, this passage discloses an aspect of <u>recording an event</u>, which is <u>not a disclosure of an employment of an emulated event handler</u>. As such, this passage does not disclose matching a member of a collection of emulated event handlers to a given event.

FIG. 2 is a representation of a graphical user interface (col. 3, line 23), that includes a function section 204 (col. 5, line 30), a recording options section 202 (col. 5, line 22 – 23), and a playback speed section 220 (col. 8, lines 49 – 50).

Recording options section 202 provides a list of selectable event type to be recorded (col. 5, lines 22 – 24). Whereas the events of recording options section 202 are to be recorded, they are <u>not yet recorded, and so, cannot be matched to anything</u>. As such, recording options section 202 is not a disclosure of matching a member of a collection of emulated event handlers to a given event.

Moreover, function section 204 includes a load button 216 and a run button 212. The Schumacher et al. patent states, at col. 5, lines 45 – 48:

> When a recorded sequence of events is desired for playback at a later time, applet event recorder 100 retrieves the recorded events via Load button 216 and initiates the playback with Run button 212.

Thus, the Schumacher et al. patent expressly describes playback as involving operation of load button 216 and run button 212. As such, in the Schumacher et al. patent, <u>a user affirmatively loads and runs a recorded event</u>. Consequently, in the Schumacher et al. patent <u>there is no need for matching a member of a collection of emulated event handlers to a given event</u>, and therefore the Schumacher et al. patent does not disclose executing the task on a selected batch of file records by matching a member of a collection of emulated event handlers to a given event.

Whereas none of the cited passages of the Schumacher et al. patent disclose matching a member of a collection of emulated event handlers to a given event, and whereas to the contrary, the Schumacher et al. patent expressly describes that **a user affirmatively loads and runs a recorded event**, and so **there is no need for matching a member of a collection of emulated event handlers to a given event**, the Schumacher et al. patent does not disclose executing a task on a selected batch of file records by matching a member of a collection of emulated event handlers to a given event, as recited in claim 1

The Dixon et al. patent does not even mention "emulation". As such, the Dixon et al. patent does not make up for this deficiency on the part of the Schumacher et al. patent.

The Office Action, on page 4, recognizes that the Schumacher et al. patent does not explicitly teach identifying an occurrence of an event that occurs, and wherein a response is performed by a human operator. However, the Office Action suggests that the Dixon et al. patent discloses this feature. In support of this position, the Office Action states that the Dixon et al. patent (i) discloses that a plurality of events is received through a computer communication interface, and cites (a) col. 14, lines 41 – 56, (b) col. 3, lines 30

– 38, and (c) the Abstract, and (ii) discloses that a response to an event is performed by a user, and cites (d) col. 77, lines 1 – 5, and (e) col. 76, lines 42 – 67.

The passage at col. 14, lines 41 – 56 explains that a frame sequence 310 is translated into an event sequence 312 using a compiler, and that event sequence 312 is received by a test control unit 311and stored in preparation for execution. However, the Dixon et al. patent, at col. 14, lines 24 – 26 explains that an event sequence is a set of instructions for execution by a computer system. Event sequence 312 is not an event to which a human operator responds.

The passage at col. 3, lines 30 – 38 also describes event instructions, and specifically states:

> The detector framing node also includes a control unit to receive a plurality of event instructions from the host computer through the computer communication interface. The event instructions selectively control a radiation generation system and an image detection system. The event instructions are executed in real time and at predetermined timing intervals.

Thus, the passage at col. 3, lines 30 – 38 explains that event instructions selectively control a radiation generation system and an image detection system. As such, this passage does not disclose an event to which a human operator responds.

The Abstract states that a detector framing node is programmable to control events according to an event instruction sequence, to control the generation and reception of image data. The Abstract does not disclose an event to which a human operator responds.

The passage at col. 77, lines 1 – 5, when read in context with lines 6 – 14, describes a flag in an event queue that notifies a user that an initialization process is finished so that the user can perform an action. However, the passage does not indicate that the user's reply is recorded, and more relevantly, the passage does not indicate that the user's reply is further employed in an emulated event handler. Instead, the purpose of having the user

reply to the event queue is <u>to synchronize operations of the queue and the user</u> (col. 77, lines 13-14).

The passage at col. 76, lines 42 – 67 prefaces the aforementioned passage at col. 77, lines 1 – 5. It indicates that a flag is used to tell a user that an event has happened (col. 77, lines 61 – 62), but the passage does not indicate that the user's reply is recorded, and more relevantly, the passage <u>does not indicate that the user's reply is further employed in an emulated event handler.</u>

Thus, in the Dixon et al. patent, **an event sequence is a set of instructions that enables a computer to selectively control a radiation generation system and an image detection system.** None of the passages cited in the Dixon et al. patent, nor any other passage in the Dixon et al. patent, discloses that a user's response to an event is of a type that is recorded and further employed in **an emulated event handler.** To the contrary, the Dixon et al. patent expressly states that the event queue and the user's reply to a flag in the event queue are **for synchronizing operations of the queue and the user.** Thus, the events and the responses disclosed by the Dixon et al. patent are not for employment in emulated event handlers, as recited in claim 1.

Moreover, whereas in the Dixon et al. patent the event sequence is a set of instructions that enables a computer to selectively control a radiation generation system and an image detection system, and whereas the Dixon et al. patent does not indicate that the user's reply is recorded, and instead expressly states that the user's reply is for synchronizing operations of the queue and the user, **the events and the event queue in the Dixon et al. patent are not similar to the events contemplated by the Schumacher et al. patent** FIG. 2, item 202. Therefore, **neither of the Dixon et al. patent nor the Schumacher et al. patent provide any motive for being combined with one another.** Thus, the cited combination of the Schumacher et al. and Dixon et al. patents is inappropriate for purposes of a section 103(a) rejection.

For the several reasons provided above, Applicants submit that claim 1 is patentable over the cited combination of the Schumacher et al. and Dixon patents.

Claims 2, 3, 6, 7, 9 and 10 depend from claim 1. At least because of this dependence, claims 2, 3, 6, 7, 9 and 10 are also patentable over the cited combination of the Schumacher et al. and Dixon patents.

Claim 11 is an independent method claim. The method includes, *inter alia*, (a) identifying an occurrence of an event that occurs while a task is being accomplished to revise a file record, and (b) recording in a memory, a response to said event, wherein said response is performed by a human operator interacting with a graphical user interface. Thus, claim 11 includes recitals similar to some of those of claim 1, as described above. As such, claim 11 is patentable over the cited combination of references for reasoning similar to that of claim 1, as it applies to these recitals.

Claims 12 through 15 depend from claim 11. Therefore, claims 12 through 15 are patentable over the cited combination of references, at least because of this dependence.

Furthermore, claim 14, in addition to depending from claim 11, includes a recital of matching a member of a collection of emulated event handlers to an event that occurs while performing a task for an obtained record. As explained above in support of claim 1, neither of the cited references discloses matching a member of a collection of emulated event handlers to an event. Therefore, claim 14, is patentable over the cited combination of references not only because claim 14 depends from claim 11, but also on its own merits.

Claim 16 is an independent system claim. The system includes a computer that executes processes of, *inter alia*, (a) identifying an occurrence of an event that occurs while a task is being accomplished to revise a file record, and (b) recording in a memory, a response to the event, wherein the response is performed by a human operator interacting with a graphical user interface. Thus, claim 16 includes recitals similar to

16

some of those of claim 1, as described above. As such, claim 16 is patentable over the cited combination of references for reasoning similar to that of claim 1, as it applies to these recitals.

Claims 17 through 20 depend from claim 16. Therefore, claims 17 through 20 are patentable over the cited combination of references, at least because of this dependence.

Furthermore, claim 19, in addition to depending from claim 16, includes a recital of matching a member of a collection of emulated event handlers to an event that occurs while performing a task for an obtained record. As explained above in support of claim 1, neither of the cited references discloses matching a member of a collection of emulated event handlers to an event. Therefore, claim 19, is patentable over the cited combination of references not only because claim 19 depends from claim 16, but also on its own merits.

Claim 21 is an independent claim that provides for a program. The program includes a process that controls a computer for, *inter alia*, (a) identifying an occurrence of an event that occurs while a task is being accomplished to revise a file record, and (b) recording in a memory, a response to the event, wherein the response is performed by a human operator interacting with a graphical user interface. Thus, claim 21 includes recitals similar to some of those of claim 1, as described above. As such, claim 21 is patentable over the cited combination of references for reasoning similar to that of claim 1, as it applies to these recitals.

Claims 22 through 25 depend from claim 21. Therefore, claims 22 through 25 are patentable over the cited combination of references, at least because of this dependence.

Furthermore, claim 24, in addition to depending from claim 21, includes a recital of matching a member of a collection of emulated event handlers to an event that occurs while performing a task for an obtained record. As explained above in support of claim 1, neither of the cited references discloses matching a member of a collection of emulated

event handlers to an event. Therefore, claim 24, is patentable over the cited combination of references not only because claim 24 depends from claim 21, but also on its own merits.

Applicants respectfully request reconsideration and withdrawal of the section 103(a) rejection of claims 1 through 3, 6, 7 and 9 through 25.

Applicants added claims 26 through 33 to even further provide the claim coverage that Applicants appear to deserve based on the prior art that was cited by the Examiner. Aspects of these claims are described, for example, at page 7, line 27 through page 8, line 14, and further represented in Table 1 and FIG. 5. A favorable consideration that also results in the allowance of claims 26 through 33 is earnestly solicited.

In view of the foregoing, Applicants respectfully submit that all claims presented in this application patentably distinguish over the prior art. Accordingly, Applicants respectfully request favorable consideration and that this application be passed to allowance.

Respectfully submitted,

April 7, 2005
Date

Charles N.J. Ruggiero, Esq.
Reg. No. 28,468
Attorney for the Applicants
Ohlandt, Greeley, Ruggiero & Perle, L.L.P.
One Landmark Square, 10th Floor
Stamford, CT 06901-2682
Tel: 203-327-4500
Fax: 203-327-6401